

Subscribe to  Updates

code.makery (/)

Learning how to code.

[Home \(/\)](#)

[Library \(/library/\)](#)

[Paths \(/paths/\)](#)

[Blog \(/blog/\)](#)

[About \(/about/\)](#)

JavaFX Dialogs (official)

Oct 28, 2014 • [Java \(/blog/tag/java/\)](#), [JavaFX \(/blog/tag/javafx/\)](#)

JavaFX 8u40 finally includes simple Dialogs and Alerts! I've been waiting for this since 2012! In the meantime I wrote about how to use Dialogs in JavaFX 2 ([/blog/javafx-2-dialogs/](#)) and later in JavaFX 8 with ControlsFX ([/blog/javafx-8-dialogs/](#)).

Now that they are available in the official JDK, let's learn how to use them.

Prerequisites

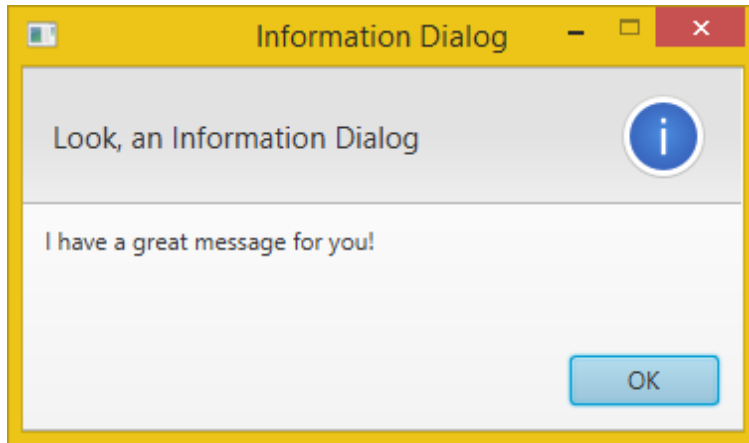
To use the official JavaFX Dialogs you need JDK 8u40 or later.

→ [Download JDK 8u40](#)

(<http://www.oracle.com/technetwork/java/javase/downloads/index.html>) or newer.

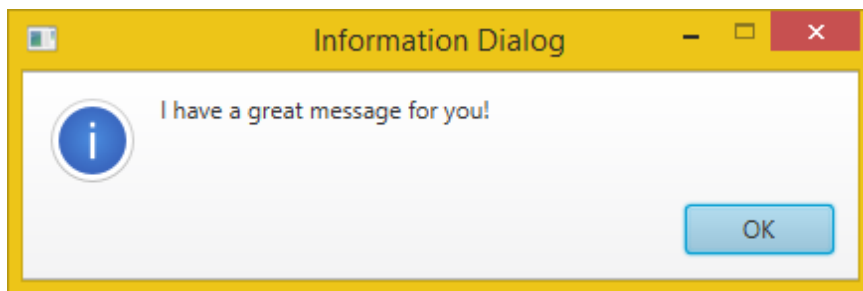
Standard Dialogs

Information Dialog



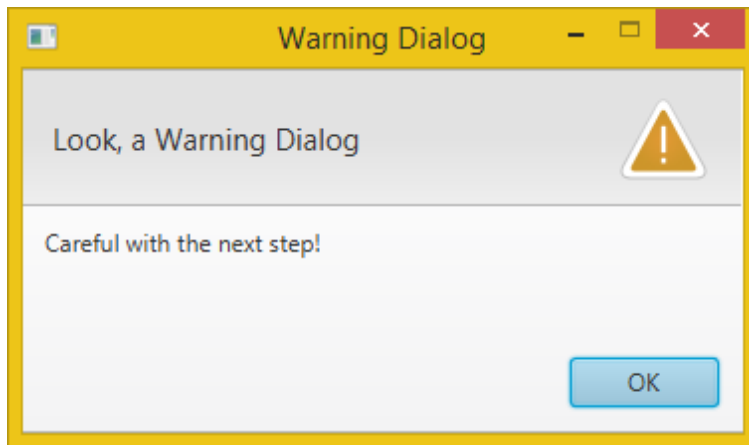
```
Alert alert = new Alert(AlertType.INFORMATION);  
alert.setTitle("Information Dialog");  
alert.setHeaderText("Look, an Information Dialog");  
alert.setContentText("I have a great message for you!");  
  
alert.showAndWait();
```

Without Header Text



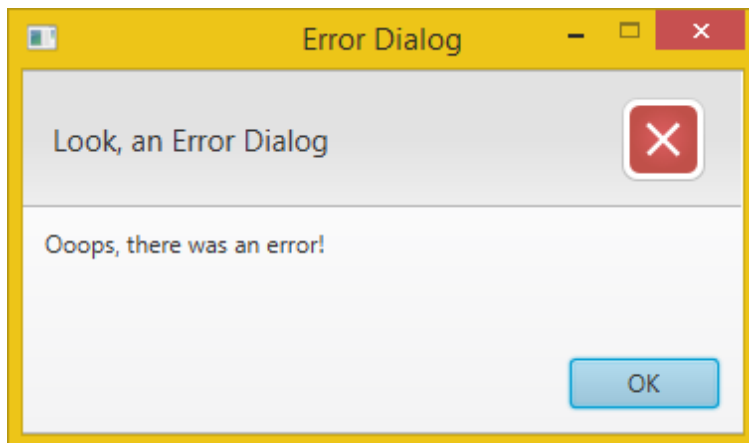
```
Alert alert = new Alert(AlertType.INFORMATION);  
alert.setTitle("Information Dialog");  
alert.setHeaderText(null);  
alert.setContentText("I have a great message for you!");  
  
alert.showAndWait();
```

Warning Dialog



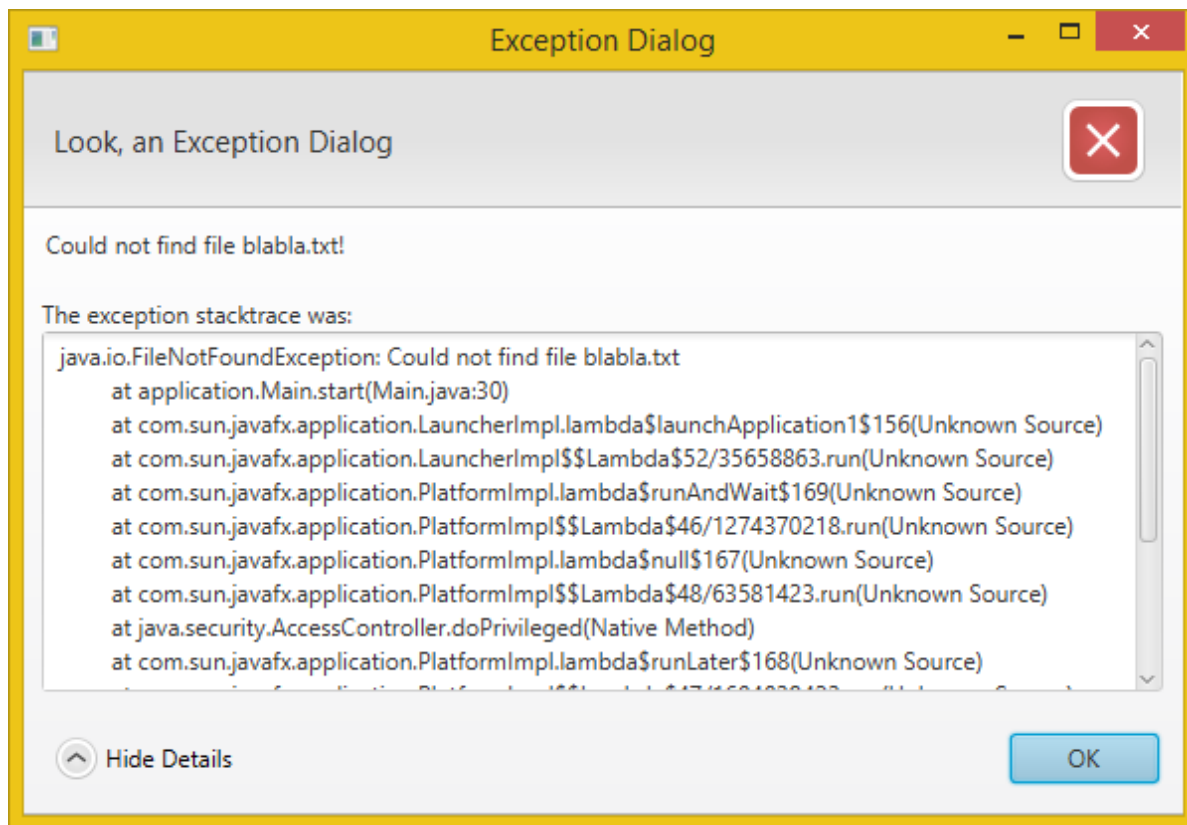
```
Alert alert = new Alert(AlertType.WARNING);  
alert.setTitle("Warning Dialog");  
alert.setHeaderText("Look, a Warning Dialog");  
alert.setContentText("Careful with the next step!");  
  
alert.showAndWait();
```

Error Dialog



```
Alert alert = new Alert(AlertType.ERROR);  
alert.setTitle("Error Dialog");  
alert.setHeaderText("Look, an Error Dialog");  
alert.setContentText("Ooops, there was an error!");  
  
alert.showAndWait();
```

Exception Dialog



There is not a complete Exception Dialog out of the box. But we can easily provide `TextArea` as expandable content.

```
Alert alert = new Alert(AlertType.ERROR);
alert.setTitle("Exception Dialog");
alert.setHeaderText("Look, an Exception Dialog");
alert.setContentText("Could not find file blabla.txt!");

Exception ex = new FileNotFoundException("Could not find file blabla.txt");

// Create expandable Exception.
StringWriter sw = new StringWriter();
PrintWriter pw = new PrintWriter(sw);
ex.printStackTrace(pw);
String exceptionText = sw.toString();

Label label = new Label("The exception stacktrace was:");

TextArea textArea = new TextArea(exceptionText);
textArea.setEditable(false);
textArea.setWrapText(true);

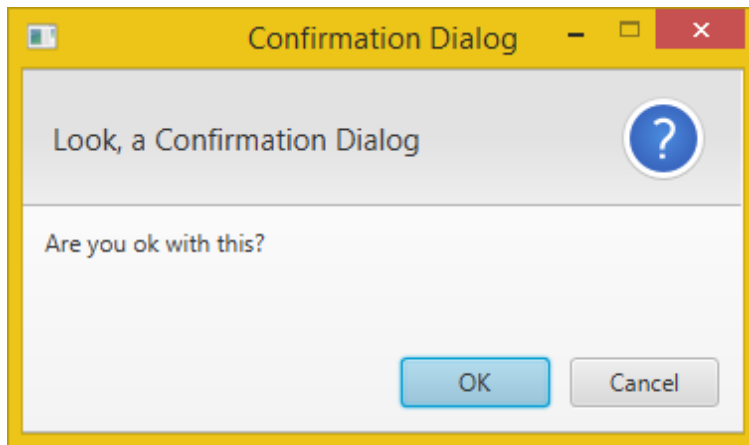
textArea.setMaxWidth(Double.MAX_VALUE);
textArea.setMaxHeight(Double.MAX_VALUE);
GridPane.setVgrow(textArea, Priority.ALWAYS);
GridPane.setHgrow(textArea, Priority.ALWAYS);

GridPane expContent = new GridPane();
expContent.setMaxWidth(Double.MAX_VALUE);
expContent.add(label, 0, 0);
expContent.add(textArea, 0, 1);

// Set expandable Exception into the dialog pane.
alert.getDialogPane().setExpandableContent(expContent);

alert.showAndWait();
```

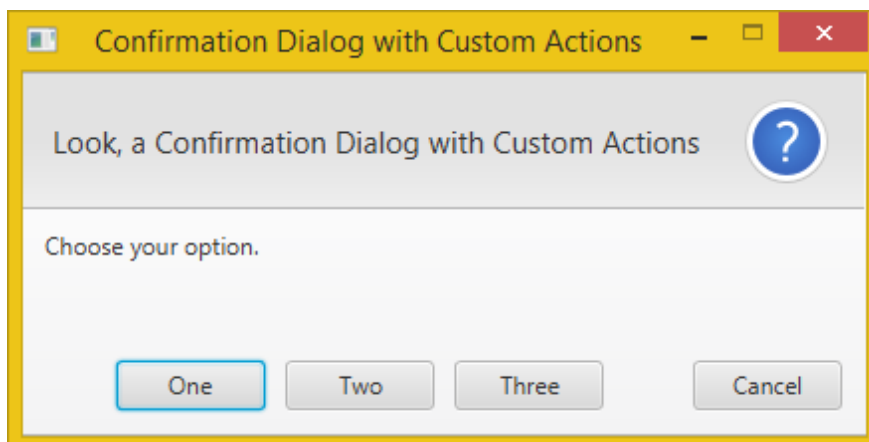
Confirmation Dialog



```
Alert alert = new Alert(AlertType.CONFIRMATION);
alert.setTitle("Confirmation Dialog");
alert.setHeaderText("Look, a Confirmation Dialog");
alert.setContentText("Are you ok with this?");
```

```
Optional<ButtonType> result = alert.showAndWait();
if (result.get() == ButtonType.OK){
    // ... user chose OK
} else {
    // ... user chose CANCEL or closed the dialog
}
```

Confirmation Dialog with Custom Actions



```
Alert alert = new Alert(AlertType.CONFIRMATION);
alert.setTitle("Confirmation Dialog with Custom Actions");
alert.setHeaderText("Look, a Confirmation Dialog with Custom Actions");
alert.setContentText("Choose your option.");

ButtonType buttonTypeOne = new ButtonType("One");
ButtonType buttonTypeTwo = new ButtonType("Two");
ButtonType buttonTypeThree = new ButtonType("Three");
ButtonType buttonTypeCancel = new ButtonType("Cancel", ButtonData.CANCEL_CLOSE);

alert.getButtonTypes().setAll(buttonTypeOne, buttonTypeTwo, buttonTypeThree, buttonTypeCancel);

Optional<ButtonType> result = alert.showAndWait();
if (result.get() == buttonTypeOne){
    // ... user chose "One"
} else if (result.get() == buttonTypeTwo) {
    // ... user chose "Two"
} else if (result.get() == buttonTypeThree) {
    // ... user chose "Three"
} else {
    // ... user chose CANCEL or closed the dialog
}
```

Text Input Dialog



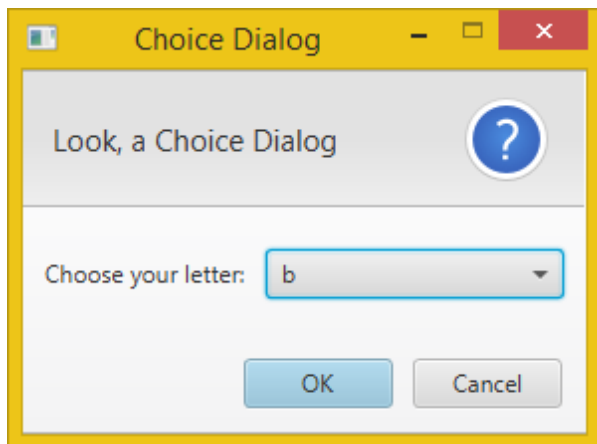
```
TextInputDialog dialog = new TextInputDialog("walter");
dialog.setTitle("Text Input Dialog");
dialog.setHeaderText("Look, a Text Input Dialog");
dialog.setContentText("Please enter your name:");

// Traditional way to get the response value.
Optional<String> result = dialog.showAndWait();
if (result.isPresent()){
    System.out.println("Your name: " + result.get());
}

// The Java 8 way to get the response value (with lambda expression).
result.ifPresent(name -> System.out.println("Your name: " + name));
```

Note: The `result.isPresent()` will return `false` if the user cancelled the dialog.

Choice Dialog




```
List<String> choices = new ArrayList<>();
choices.add("a");
choices.add("b");
choices.add("c");

ChoiceDialog<String> dialog = new ChoiceDialog<>("b", choices);
dialog.setTitle("Choice Dialog");
dialog.setHeaderText("Look, a Choice Dialog");
dialog.setContentText("Choose your letter:");

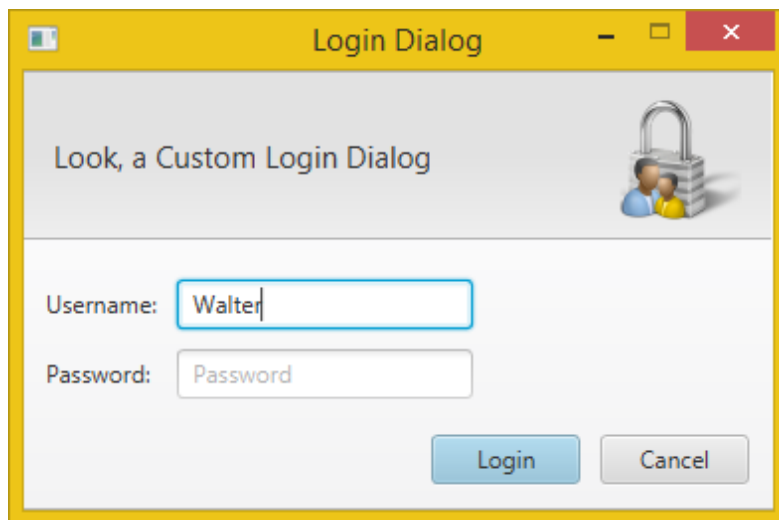
// Traditional way to get the response value.
Optional<String> result = dialog.showAndWait();
if (result.isPresent()){
    System.out.println("Your choice: " + result.get());
}

// The Java 8 way to get the response value (with lambda expression).
result.ifPresent(letter -> System.out.println("Your choice: " + letter));
```

Note: The `result.isPresent()` will return `false` if the user didn't choose anything or cancelled the dialog.

Custom Login Dialog

Here is an example of how to create a custom dialog with a login form:



```
// Create the custom dialog.
Dialog<Pair<String, String>> dialog = new Dialog<>();
dialog.setTitle("Login Dialog");
dialog.setHeaderText("Look, a Custom Login Dialog");

// Set the icon (must be included in the project).
dialog.setGraphic(new ImageView(this.getClass().getResource("login.png").toString()));

// Set the button types.
ButtonType loginButtonType = new ButtonType("Login", ButtonData.OK_DONE);
dialog.getDialogPane().getButtonTypes().addAll(loginButtonType, ButtonType.CANCEL);

// Create the username and password labels and fields.
GridPane grid = new GridPane();
grid.setHgap(10);
grid.setVgap(10);
grid.setPadding(new Insets(20, 150, 10, 10));

TextField username = new TextField();
username.setPromptText("Username");
PasswordField password = new PasswordField();
password.setPromptText("Password");

grid.add(new Label("Username:"), 0, 0);
grid.add(username, 1, 0);
grid.add(new Label("Password:"), 0, 1);
grid.add(password, 1, 1);

// Enable/Disable login button depending on whether a username was entered.
Node loginButton = dialog.getDialogPane().lookupButton(loginButtonType);
loginButton.setDisable(true);

// Do some validation (using the Java 8 lambda syntax).
username.textProperty().addListener((observable, oldValue, newValue) -> {
    loginButton.setDisable(newValue.trim().isEmpty());
});

dialog.getDialogPane().setContent(grid);

// Request focus on the username field by default.
Platform.runLater(() -> username.requestFocus());

// Convert the result to a username-password-pair when the login button is clicked.
```

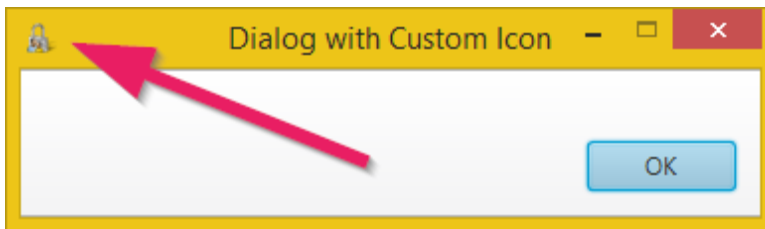
```
dialog.setResultConverter(dialogButton -> {
    if (dialogButton == loginButtonType) {
        return new Pair<>(username.getText(), password.getText());
    }
    return null;
});

Optional<Pair<String, String>> result = dialog.showAndWait();

result.ifPresent(usernamePassword -> {
    System.out.println("Username=" + usernamePassword.getKey() + ", Password=" + usernamePassw
});
```

Styling the Dialogs

Custom Icon



In the current version it's a bit cumbersome to get to the Dialog's `Stage` to be able to set its icon. Here is how:

```
// Get the Stage.
Stage stage = (Stage) dialog.getDialogPane().getScene().getWindow();

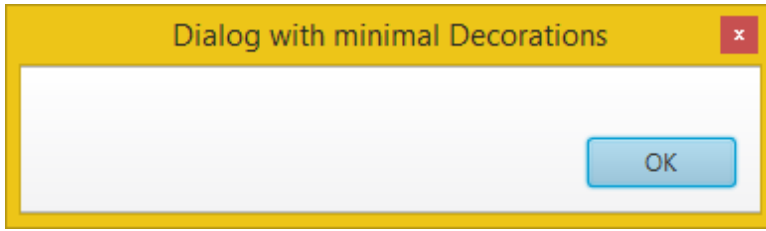
// Add a custom icon.
stage.getIcons().add(new Image(this.getClass().getResource("login.png").toString()));
```

According to this bug report (<https://javafx-jira.kenai.com/browse/RT-38895>) the final version of the JavaFX 8u40 Dialogs should use the same icon as the application that it is running from. In that case you would need to set its owner and the Dialog would get the owner's icon:

```
dialog.initOwner(otherStage);
```

Minimal Decorations

Another option is to remove the icon and use only minimal window decorations.



```
dialog.initStyle(StageStyle.UTILITY);
```

Other Options

Setting the Owner

You can specify the owner `Window` for a dialog. If no owner or null is specified for the owner, it is a top-level, unowned dialog.

```
dialog.initOwner(parentWindow);
```

Setting the Modality

You can specify the modality for a dialog. The modality must be one of `Modality.NONE`, `Modality.WINDOW_MODAL`, or `Modality.APPLICATION_MODAL`.

```
dialog.initModality(Modality.NONE);
```

API Documentation

For more information on the Dialogs have a look at the JavaFX API docs:

- `Alert` (<http://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/Alert.html>)
- `Dialog` (<http://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/Dialog.html>)
- `TextInputDialog`
(<http://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/TextInputDialog.html>)
- `ChoiceDialog`
(<http://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/ChoiceDialog.html>)
- `DialogPane`
(<http://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/DialogPane.html>)

Comments

Sponsored

Man Finds Weird Object In Garden- When Police Arrive They Say: You Shouldn't Have Seen This

People Today

Faith Hill's Daughter Is Probably The Most Stunning Ever

Noteably

She Was Beautiful, Now It's Hard To Look At Her

Smartingly

She Is One Of The Largest Athletes In The World - And She Is Truly Massive

TheDaddest

[Pics] Everyone Laughed At Her At The Wedding, But 6 Years Later She Shows Her Transformation

Petdiver

Wardrobe Malfunctions That Actors Still Regret

ItsTheVibe

226 Comments

 Login ▼

G

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS  51

Share

Best Newest Oldest**Perla Z**

9 years ago

This is awesome! Thank you!

8  Reply • Share ›**Marco Jakob** Mod Perla Z

9 years ago

Wow, you are fast with writing comments... I've just published the article.

5  Reply • Share ›**Perla Z** Marco Jakob

9 years ago

I was so waiting for it

  Reply • Share ›**Marco Jakob** Mod Perla Z

9 years ago

Waiting for the blog post or waiting for Oracle to finally put the dialogs in the official JDK? :-)

  Reply • Share ›**Perla Z** Marco Jakob

9 years ago

both! :D

51 o Reply ● Share ›



Avata

This comment was deleted.

**Sajib Srs**

➔ Guest



8 years ago

Smooth. Who said coders are not romantic ;)

3 o Reply ● Share ›

**Marco Jakob** Mod

➔ Sajib Srs

8 years ago

:-)

2 o Reply ● Share ›

**Aurélien**

9 years ago

Thanks for your work. Its a great post, it will help me a lot.

1 o Reply ● Share ›

M**Michael Ramirez**

9 years ago

If I were to create a "Save" dialog. How would I prevent the "Save" button from closing the dialog until the save operation was completed.

o o Reply ● Share ›

**Marco Jakob** Mod

➔ Michael Ramirez



9 years ago edited

I think they are meant for simple dialogs with only limited interaction. It might be possible to not return immediately when the "Save" button is clicked but maybe there are better ways:

- * Why does the dialog need to stay open? If it's a long running operation you could close the dialog and then show some progress bar.
- * If you need more functionality then you might be better off by just implementing your own Stage.
- * If you still want to use the dialogs, you might need to extend the Dialog class. Have a look at the Dialogs source here: <https://bitbucket.org/contr...>

1 o Reply • Share ›

**Marco Jakob** Mod

➔ Michael Ramirez



9 years ago edited

Actually, I found a bug report for this that also contains a workaround:

<https://javafx-jira.kenai.c...>

Here is the workaround from the issue tracker:

Now I found the following workaround by performing a lookup of the OK button and using an action filter which consumes the button action event in case of validation errors:

```
final DialogPane dlgPane = dlg.getDialogPane();
dlgPane.getButtonTypes().add(ButtonType.OK);
dlgPane.getButtonTypes().add(ButtonType.CANCEL);

final Button btOk = (Button) dlg.getDialogPane().lookupB
btOk.addEventFilter(ActionEvent.ACTION, (event) -> {
    if (!validateAndStore()) {
        event.consume();
    }
});

dlg.showAndWait();
```

o o Reply • Share ›

N

Nick

9 years ago

Is there a way to be able to change the icon of the window?
like for the Stage class : `getIcons().add(Image)`

1 o Reply • Share ›

**Marco Jakob** Mod

➔ Nick



9 years ago edited

I've found a way to change the icon:

```
Stage stage = (Stage) dialog.getDialogPane().getScene().
stage.getIcons().add(new Image(this.getClass().getResour
```

You could also remove the icon with:

```
dialog.initStyle(StageStyle.UTILITY);
```

In the final version the dialog should have the same icon as its owner. You can set the owner window like this.

```
dialog.initOwner(otherStage);
```

I don't know if this already works now, because there was a bug report that was fixed about a day ago. See <https://javafx-jira.kenai.c...>

o o Reply • Share ›



Marco Jakob Mod

➔ Marco Jakob

9 years ago

I've added the styling and icon examples to the blog post above.

o o Reply • Share ›

N

Nick ➔ Marco Jakob

9 years ago

Thank you very much! :)

o o Reply • Share ›



Claudio Mezzasalma

➔ Marco Jakob

9 years ago

It seems that `initModality` is not working for me... I don't have a snippet right now but simply the window below keeps receiving events regardless of the value set... Any clue?

o o Reply • Share ›



Marco Jakob Mod

➔ Claudio Mezzasalma

9 years ago

If you think it might be a bug, please report it to: <https://javafx-jira.kenai.com/>

1 o Reply • Share ›



Claudio Mezzasalma

➔ Marco Jakob

9 years ago

Honestly it's a quite border line scenario since I'm opening my dialog from a drag and drop event fired inside a pane which is inside a JFXPanel and used in NetBeans Platform... I'll do some research and try to replicate it with a straight JavaFX App.

o o Reply • Share ›

I

ilxan

9 years ago

I'm trying to change the width and height of a TextInputDialog by using the setHeight() / setWidth() methods. Unfortunately its not working. Here is a snippet, what am i doing wrong?

```
TextInputDialog dialog = new TextInputDialog();
```

```
dialog.initOwner(stage);
dialog.setHeight(300.0);
dialog.setWidth(600.0);
dialog.setTitle("Adding new selector");
dialog.setHeaderText("test");
dialog.setContentText("text:");
```

```
Optional<string> result = dialog.showAndWait();
```

o o Reply • Share ›

**Marco Jakob** Mod

➔ ilxan

9 years ago

I don't know if it works, but you could try:

```
dialog.getDialogPane().setHeight(...);
dialog.getDialogPane().setWidth(...);
```

o o Reply • Share ›

I

ilxan ➔ Marco Jakob

9 years ago

Thanks Marco,

i got it to work by setting the style:

```
dialog.getDialogPane().setStyle("-fx-max-width:600px; -fx-max-height: 300px; -fx-pref-width: 600px; -fx-pref-height: 300px;");
```

o o Reply • Share ›

HS **Hoi Sidwell**

9 years ago

Hi, excellent article. I have it bookmarked.

There is probably no way of getting rid of that last pesky x icon on the minimal decorations, I suppose, shy of an UNDECORATED stage altogether? css or some other hack perhaps?

Also, I am not seeing a way to position the dialog. Any tips?

o o Reply ● Share >



Marco Jakob **Mod**

→ Hoi Sidwell

9 years ago

There is actually an undecorated way:
`dialog.initStyle(StageStyle.UNDECORATED);`
 Also see here: <http://stackoverflow.com/qu...> and here:
<http://pixelduke.wordpress....>

About positioning: I think the dialogs should automatically be centered relative the their owner but I might be wrong. Try setting the owner and then using `dialog.setX(...)` and `dialog.setY(...)`;

o o Reply ● Share >



driv3l

9 years ago

What's the best way to handle localization with these new dialogs? The problem I am having is with replacing the localized text in the default dialog buttons. I basically just want to replace the text (or even the buttons) with my own localized versions. It seems the dialogs won't work if you remove the default buttons and add your own e.g. `TextInputDialog`. I already have a keyed bundle that I am using and just want to pass my "ok" and "cancel" to the buttons.

o o Reply ● Share >



Marco Jakob **Mod**

→ driv3l

9 years ago

Using your own buttons should work. Just make sure you use the `ok_done` and `cancel_close` `ButtonData`:

```
ButtonType buttonTypeCancel = new ButtonType("Cancel", B
ButtonType buttonTypeOk = new ButtonType("OK", ButtonDat

dialog.getButtonTypes().setAll(buttonTypeCancel, buttonT
```



o o Reply • Share ›



driv3l

➔ Marco Jakob



9 years ago

Hi Marco,

Thanks for the prompt response. Unfortunately it doesn't work with a TextInputDialog.

I believe it may be a bug. If you look at the source code for TextInputDialog:

<https://bitbucket.org/contr...>

Instead of looking at the ButtonData to see if it's OK_DONE, it's actually checking to see if the button pressed is it's own reference for OK.

o o Reply • Share ›



Marco Jakob Mod

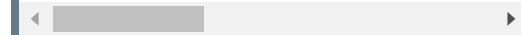


➔ driv3l

9 years ago

I don't know what their idea is about how this should be done properly. But you could probably set your own result converter:

```
dialog.setResultConverter((dialogButt
```



o o Reply • Share ›



driv3l

➔ Marco Jakob



9 years ago

Thanks Marco.. that fixes it.

I hope they decide to change it so that it actually looks at the ButtonData and not the reference. IMO, that would be the right way to handle it.

o o Reply • Share ›



Marco Jakob Mod



➔ driv3l

9 years ago

I've left a comment on the page of the dialog designers and hope they will see this discussion and maybe join in:

<http://fxexperience.com/201...>

o o Reply • Share ›



Jonathan Giles

➔ driv3l

9 years ago

I filed RT-39330 and fixed the issue here: <https://javafx-jira.kenai.c...>

Please, in the future file bugs in the JavaFX jira issue tracker. Thanks!

23 o o Reply • Share ›

J

JC

9 years ago

Is there a way of setting the parent stage? Like the old versions?

o o Reply • Share ›

W

William Vincent

➔ JC

9 years ago

Dialog.initOwner(Window wdw)

o o Reply • Share ›



Marco Jakob Mod

➔ William Vincent

9 years ago

I've added some information about setting the owner and the modality to the blog post.

o o Reply • Share ›



michaelpellis

9 years ago

When will there be SceneBuilder support for Dialogs? Or is there already a way to use Dialogs with SceneBuilder?

o o Reply • Share ›



Marco Jakob Mod

➔ michaelpellis

9 years ago

It should be coming to SceneBuilder. But I don't know

when exactly. See here: <http://openjdk.java.net/jep...>

o o Reply • Share ›



michaelpellis

➔ Marco Jakob



9 years ago

@Marco Jakob. That was a quick reply. Many thanks. I'm just starting with JavaFX and am loving it so far. At first i thought my application would not benefit too much from SceneBuilder particularly where the UI is highly dynamic, but I have found an excellent use for it where I can let system administrators of our application use SceneBuilder directly to design there own forms and then store/restore the state of these forms by trawling the Node tree for input items and saving/loading the content as Java Properties. Saves me having to roll my own form designer!

I have another query which I think is a bug (or my lack of understanding with SceneBuilder - I am unable to configure a ChoiceBox with its items in SceneBuilder - completely off topic but where should I address that query to?

Many thanks

-- Michael Ellis

1 o Reply • Share ›



Marco Jakob Mod



➔ michaelpellis

9 years ago

Hi Michael,

I also think SceneBuilder is very helpful and probably the best visual editor I've seen in my Java career. It's great that your system administrators can use it in such a way.

If you've found a bug in scene builder it's best if you report it here:

<https://javafx-jira.kenai.com/>

o o Reply • Share ›



michaelpellis





Marco Jakob

9 years ago

It looks like the limitation is known about. I added my tuppence worth here: <http://goo.gl/vV3LgX> I'm confused though as the implication from the comments that precede mine is that setting the contents for ChoiceBox and ComboBox are not considered necessary!!!!

o o Reply • Share ›

**Gazda Miloš**

9 years ago edited

Hello, good job with the article!

Is there a way how Alert could consume a KeyEvent? I have a simple login AnchorPane (username/password/login button) which method onKey(KeyEvent event) is assigned to handle released key to process the login.

```
private void initialize() {
    anchorPane.setOnKeyReleased((event) -> onKey(event))
}
```

and

```
public void onKey(KeyEvent keyEvent) {
    if (keyEvent.getCode() == KeyCode.ENTER) {
        login();
    }
}
```

[see more](#)

o o Reply • Share ›

**Marco Jakob** Mod

→ Gazda Miloš

9 years ago

Sorry, I don't know right away what you could do. But maybe it helps when you look at the source code of Alert to see what happens when you hit enter:

<https://bitbucket.org/contr...>

o o Reply • Share ›

— — — 9 years ago

is there a bug with the alert dialog? as when i set UNDECORATED and add a css border around it i get a 10px white banner on the RHS of the dialog?

0 0 Replv ● Share >



Subscribe to receive updates by Email (<https://tinyletter.com/codemakery>).

Get updates

Created by Marco Jakob (/about) • Licensed under Creative Commons Attribution 4.0
(<https://creativecommons.org/licenses/by/4.0/>)